



Diagnostic des Systèmes Mobiles Communicants

Insaf Sassi, Alexia Gouin, Jean-Marc Thiriet

► To cite this version:

Insaf Sassi, Alexia Gouin, Jean-Marc Thiriet. Diagnostic des Systèmes Mobiles Communicants. Journées Nationales des Communications Terrestres (JNCT 2015), Jun 2015, Valence, France. hal-01248330

HAL Id: hal-01248330

<https://hal.science/hal-01248330>

Submitted on 24 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diagnostic des Systèmes Mobiles Communicants

Insaf SASSI[#], Alexia GOUIN[#], Jean-Marc THIRIET[#]

[#]Gipsa-lab, Université Grenoble Alpes, France
¹insaf.sassi@gipsa-lab.grenoble-inp.fr

Résumé— Les systèmes commandés en réseau, comme les robots mobiles communicant avec une station distante, sont une nouvelle génération de systèmes distribués où la communication entre les actionneurs, les capteurs et le contrôleur est assurée par un réseau de communication. Mais l'intégration du réseau dans la boucle de commande engendre des problèmes au niveau de la performance du système d'où la nécessité d'avoir une méthode de diagnostic et de prise de décision, pour localiser les défaillances. Ce papier présente une méthode de diagnostic utilisant l'approche bayésienne.

Mots clés : systèmes commandés en réseau, robot mobile, diagnostic, réseau bayésien

I. INTRODUCTION

Les systèmes commandés en réseau NCSs (Networked Control Systems) sont des systèmes avec retour d'état où la boucle de commande intègre un réseau de communication filaire ou sans fil. Plusieurs domaines d'application tels que le transport, l'énergie, et l'environnement recourent aux systèmes commandés en réseau dans le but de répartir la tâche à accomplir ou bien pour la coopération ou le contrôle à distance. L'utilisation d'un réseau pour interconnecter les différents composants du système paraît utile vu la complexité des systèmes. La réduction de la complexité des installations, le poids et l'espace réduit des NCSs se révèlent comme des avantages pour ces systèmes [1]. Le NCS est composé principalement de quatre composants dans le but de fermer la boucle de commande :

- les capteurs qui ont pour rôle de collecter les informations d'état du sous-système à commander;
- le contrôleur dont le rôle est de fournir les commandes;
- les actionneurs, qui en recevant les commandes, fournissent un travail utile à l'exécution de tâches;
- le réseau de communication qui permet aux différents composants du NCS de communiquer.

Les informations entre le contrôleur principal et les actionneurs et entre les capteurs et le contrôleur principal sont échangées via un réseau comme le montre la figure 1.

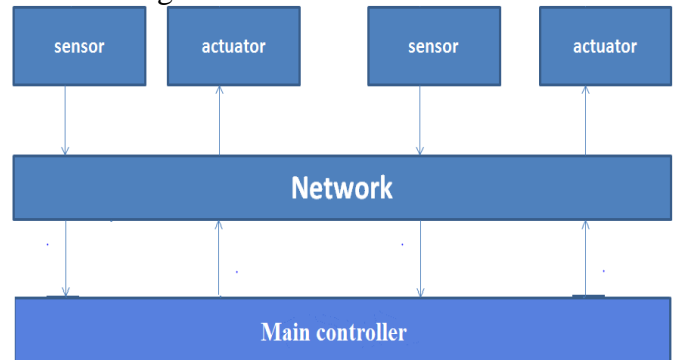


Fig. 1 : Schéma du système commandé en réseau

Le NCS peut être implémenté suivant deux configurations [2] :

- structure directe : le NCS est alors composé d'un contrôleur principal distant et de capteurs et d'actionneurs. Dans ce cas, il n'y a pas de contrôleur local;
- structure hiérarchique : elle est composée d'un contrôleur principal qui envoie, après avoir reçu les mesures de capteurs à chaque période d'échantillonnage, le signal de consigne au contrôleur local via le réseau de communication. Le contrôleur local attend le signal de consigne avant d'exécuter la boucle de commande locale. La boucle de commande principale (boucle de commande en réseau), fournissant le signal de référence, doit avoir une période d'échantillonnage plus grande que celle de la boucle locale qui possède une dynamique plus rapide.

Cependant, l'intégration d'un réseau dans la boucle de commande génère des problèmes qui modifient les performances du système. Plusieurs paramètres,

qui représentent la qualité de service QoS du réseau, ont un impact sur la performance du système commandé en réseau et sur sa qualité de commande QoC [1, 2, 3, 4, 5, 6, 7, 8, 11] :

- les pertes de paquets;
- le délai;
- la gigue;
- le débit.

L'article est organisé comme suit. L'introduction est suivie de la section 2 qui décrit l'architecture du système commandé via un réseau sans fil. Dans la section 3, une étude sur le diagnostic, les aspects théoriques des RBs et des RBs pour le diagnostic sont présentés. Dans la section 4, nous présentons notre propre réseau bayésien qui sert à diagnostiquer le dysfonctionnement du système.

II. ARCHITECTURE DU SYSTEME

Le système est composé d'un robot mobile commandé via un réseau sans fil par une station distante. On décompose le système en 3 parties [9]:

- la partie commande : elle correspond au contrôleur, calculateur des lois de commandes servant à piloter la partie opérative;
- la partie opérative : elle est composée des capteurs et des actionneurs. A chaque période d'échantillonnage, les capteurs envoient les mesures à la partie commande qui envoie, à son tour, les consignes aux actionneurs;
- la partie réseau : elle correspond au réseau de communication qui sert à l'échange d'informations entre la partie opérative et la partie commande.

La principale mission du robot est l'atteinte de la cible en évitant les obstacles. Mechraoui [4] a conçu une architecture de commande embarquée et débarquée que nous avons implémentée comme le montre la figure 2.

Le robot peut être autonome (lorsqu'il est en dehors de la zone de couverture de la station) et dans ce cas tout calcul nécessaire est effectué au niveau de son microcontrôleur embarqué. Sinon, le robot est connecté à une station distante (ordinateur) et dans ce cas une partie des calculs est effectuée au niveau de la station. Le contrôleur principal reçoit, à chaque période d'échantillonnage, les mesures des capteurs (la position et l'orientation du robot), prend des décisions pour le robot. Il calcule les lois de commandes (la vitesse linéaire et angulaire) et envoie les consignes (les vitesses angulaires gauche et droite) au contrôleur local implémenté sur le robot. Donc, on est plutôt dans une structure hiérarchique du système comme le montre la figure 2.

Dans le but de réduire les calculs et d'optimiser la consommation d'énergie au niveau du robot, le contrôleur local correspond à un algorithme de contrôle simple qui est un régulateur PI.

Le réseau sans fil est implanté entre le contrôleur principal et les actionneurs d'une part et entre les capteurs qui effectuent et le contrôleur principal d'autre part. Le réseau implanté fonctionne en mode infrastructure où la station de contrôle distante gère l'accès au réseau.

La tâche capteur est guidée par le temps, c'est-à-dire elle est exécutée périodiquement. Les tâches contrôleur et actionneur, quant à elles, sont guidées par les événements c'est-à-dire que ces deux tâches ne s'exécutent que lorsqu'elles reçoivent les échantillons provenant du réseau.

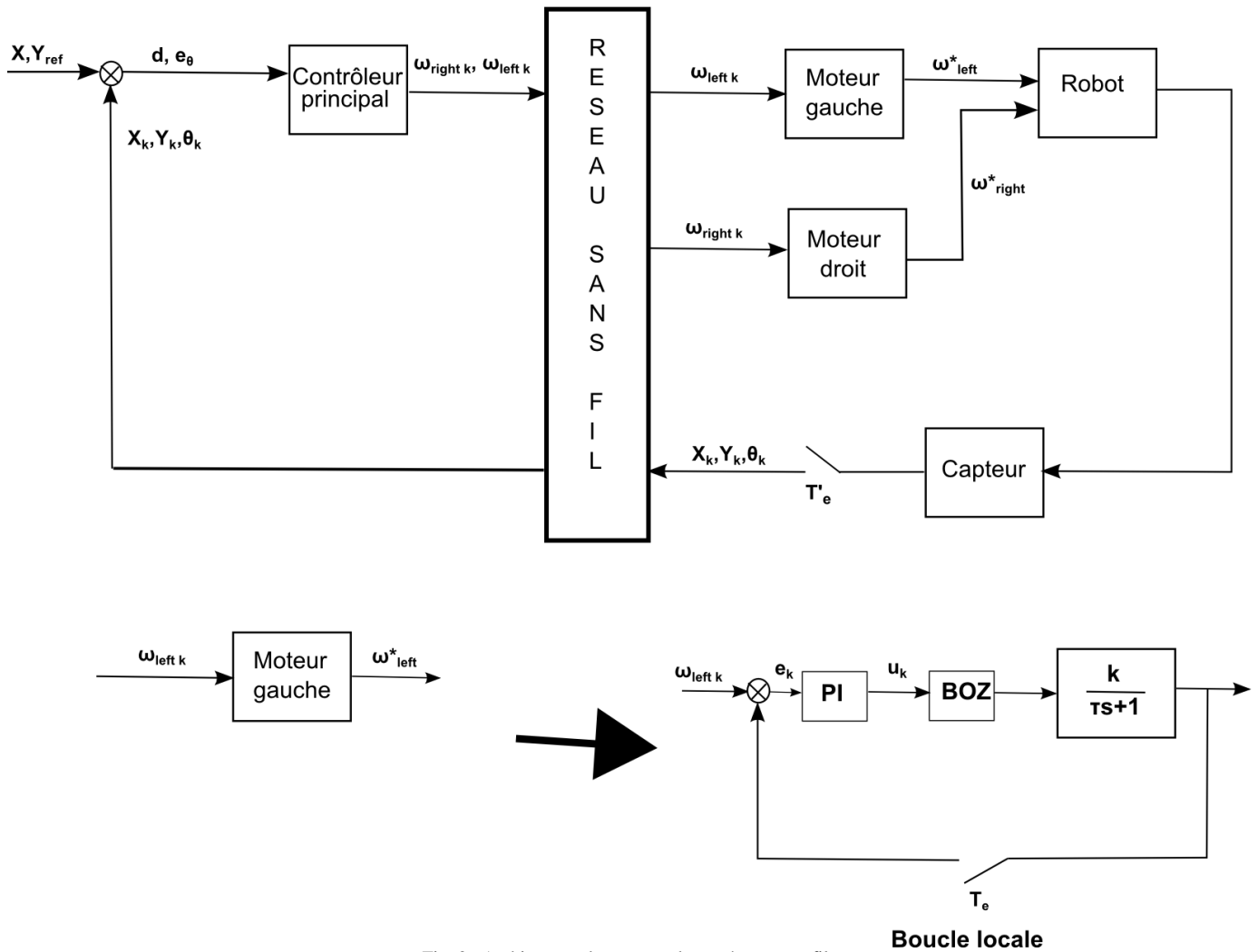


Fig. 2 : Architecture de commande en réseau sans fil

Après avoir implémenté l'architecture de commande du système, une consigne de position (cible à atteindre) $(x_d, y_d) = (1, 1)$ est appliquée. L'état initial du robot est $(x_i, y_i, \theta_i) = (0, 0, 0)$. La simulation est faite avec Matlab et TrueTime (version 2.0 bêta 5) [Cervin et al, 2010]. La nouvelle position du robot est prise à chaque période d'échantillonnage 400ms (période d'échantillonnage de la boucle principale). Les résultats de simulation sont présentés sur les figures 3 et 4.

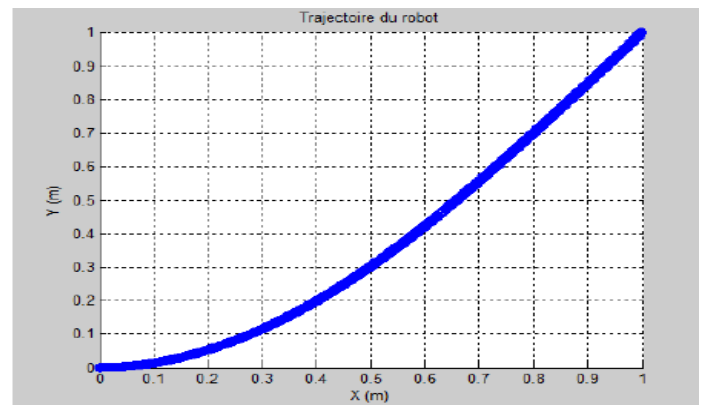


Fig. 3 : Trajectoire du robot en simulation

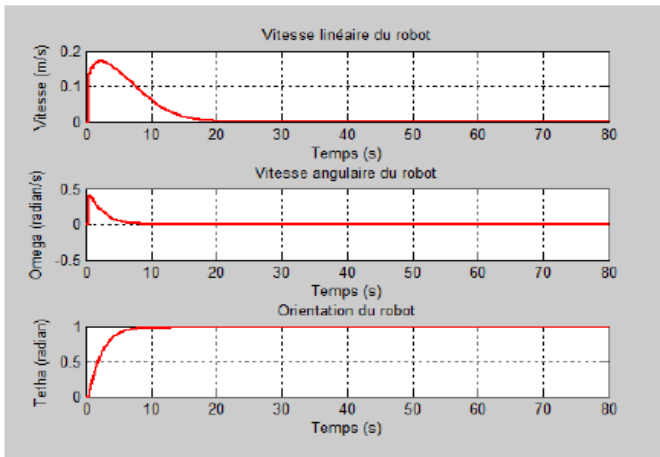


Fig. 4 : Évolution des vitesses linéaire et angulaire et orientation du robot

La figure 3 montre l'évolution de la trajectoire du robot en déplacement vers la cible. La figure 4 illustre l'évolution de la vitesse linéaire, de la vitesse angulaire et l'orientation θ du robot. L'orientation initiale du robot est nulle puis il change d'orientation afin d'atteindre la cible.

Lorsque la vitesse linéaire est strictement supérieure à 0, la trajectoire du robot évolue jusqu'à sa destination comme le montre la figure 4. Le robot rejoint sa destination après 20s où la vitesse linéaire devient nulle.

III. RESEAUX BAYESIENS POUR LE DIAGNOSTIC

A. Le diagnostic

Le diagnostic d'un système consiste à localiser une défaillance afin de prendre une décision pour remettre le système dans son état normal. Le diagnostic d'un défaut consiste à utiliser une méthode permettant de déterminer si un composant est défaillant ou non et de localiser l'origine de la défaillance à partir des observations. En outre, une défaillance est un état qui peut mener à un dysfonctionnement du système. Un défaut peut être caractérisé par une déviation, en dehors d'un intervalle acceptable, d'une variable ou d'un paramètre observé. La tâche de diagnostic est divisée en 3 sous-tâches [11] comme le montre la figure 5 :

- la détection du défaut : elle consiste à indiquer et observer un dysfonctionnement dans le système;

- la localisation du défaut : elle consiste à déterminer l'élément défaillant;
- l'identification du défaut : elle consiste à déterminer la forme et l'amplitude du défaut.

Les premières méthodes de diagnostic ont été basées sur la redondance de matériels considérés critiques pour le fonctionnement d'un système [12].

Cependant, cette méthode possède des inconvénients liés aux coûts dus à la multiplication des éléments et au poids supplémentaire. D'autres méthodes du domaine de l'automatique et de l'intelligence artificielle ont vu le jour permettant d'éliminer la redondance matérielle.

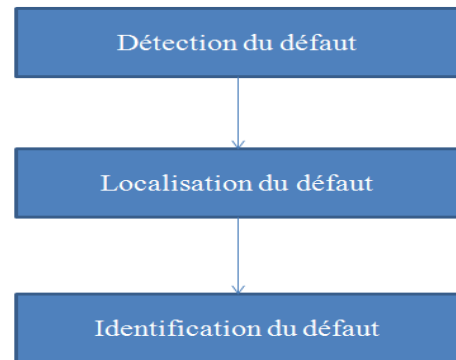


Fig. 5 : Étapes de diagnostic

En effet, les méthodes de diagnostic se partagent en deux approches [20] :

- l'approche quantitative utilisant des modèles analytiques tels que l'espace de parité, les observateurs et l'estimation de paramètres, bond graph...
- l'approche qualitative qui regroupe des méthodes basées sur les techniques de l'intelligence artificielle et qui ne font pas appel aux modèles analytiques.

Dans la première approche, il faut avoir un modèle mathématique qui représente les phénomènes physiques pour réaliser le diagnostic. Cependant, il est parfois très difficile d'avoir un modèle qui traduit fidèlement le comportement du système. En ce qui concerne la deuxième approche, elle exploite les données disponibles pour faire l'apprentissage et l'expertise du système pour construire les règles. Mais dans le cas où la base de données est incomplète ou si plusieurs cas sont indisponibles, il sera difficile d'utiliser cette approche. Dans le cas

des systèmes complexes et la disponibilité des connaissances expertes, il est possible d'utiliser les réseaux bayésiens [13].

B. RÉSEAUX BAYÉSIENS

Les réseaux bayésiens (RB) sont des modèles graphiques probabilistes utilisés dans le diagnostic utilisant l'intelligence artificielle. Ils représentent les dépendances entre événements ainsi que leurs probabilités d'apparition. Les RBs connaissent plusieurs applications telles que l'aide à la décision pour les véhicules autonomes, la modélisation des risques et surtout le diagnostic qui est l'objectif du travail. Le principal avantage que possède ce modèle est la possibilité de rassembler et de fusionner des données de diverses natures [13] : retour d'expérience, expertise (règles logiques, équations, statistiques ou probabilités) et observations. Les réseaux bayésiens, dont le nom provient des travaux de Thomas Bayes sur la théorie des probabilités, sont le résultat de recherches effectuées dans les années 1980 par J. Pearl.

Un réseau bayésien est défini par [13] comme suit :

- un graphe acyclique orienté $G = (V, E)$ où V est l'ensemble des nœuds et E est l'ensemble des arcs;
- un espace probabilisé fini (Ω, Z, P) ;
- un ensemble de variables aléatoires associées aux nœuds du graphe tel que :

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | C(V_i)) \quad \text{où}$$
 $C(V_i)$ est l'ensemble de parents de V_i

Les réseaux bayésiens constituent une approche qualitative qui fournit un modèle probabiliste pour représenter les relations de cause à effet entre différents événements dépendants. Cela revient à identifier les variables et leurs espaces d'états et à déterminer la structure du RB. Puis, à chaque nœud, on associe la table de probabilité qui le caractérise. Un nœud V_i est caractérisé par des probabilités conditionnelles s'il possède des nœuds parents. Pour les nœuds racines, qui n'ont pas de parents, on associe une loi de probabilité marginale.

L'inférence dans un réseau bayésien est un calcul de probabilités conditionnelles : elle consiste en une mise à jour des probabilités des variables non

observées après observation des valeurs d'un certain nombre d'autres variables. Le problème général pour calculer les probabilités à posteriori dans un réseau bayésien est NP-difficile. Mais il existe des algorithmes pour résoudre ce problème tel que celui de Pearl et l'algorithme de l'arbre de jonction qui a été développé par Jensen.

C. RESEAUX BAYESIENS POUR LE DIAGNOSTIC

Plusieurs modèles de réseaux bayésiens ont été proposés pour faire du diagnostic. Pour les systèmes dont la durée de vie est aléatoire, le RB, qui est une approche probabiliste, est une bonne méthode pour évaluer l'état du système. Le réseau bayésien est un bon formalisme de modélisation dans l'incertain [14]. Un autre avantage est la possibilité de représenter les dépendances entre les défaillances. Bottonne et al [15] proposent, dans son modèle pour le diagnostic d'un satellite, 2 types de nœuds : un nœud mesurable qui correspond à l'état de paramètres mesurables et un nœud non mesurable qui représente un défaut d'un composant ou de tout le système. Un autre type de nœud peut être utilisé dans le modèle, il représente une variable de l'environnement dans lequel le système fonctionne. Le diagnostic dans ce type de modèle consiste à déterminer la probabilité marginale maximale des nœuds non mesurables sachant l'état de nœuds mesurables, ce qui donne comme résultat la cause la plus probable de la défaillance du système. Cheng et al [16] ont proposé un réseau bayésien dynamique pour la maintenance. Leur objectif est de diagnostiquer la cause de l'arrêt d'un train à grande vitesse. Dans leur modèle, ils utilisent un nœud *monitor* qui a pour rôle la surveillance. Quand le nœud *monitor* découvre une défaillance au niveau du système, il communique l'information au nœud *maintenance* (la probabilité de la maintenance augmente) pour décider d'effectuer ou non une action de maintenance. Le nœud *maintenance* agit sur le système *Automatic train protection (ATP)*, qui aide à éviter les collisions, et améliore l'état du nœud *ATPCU* à l'instant suivant. Un autre modèle bayésien a été développé par une équipe de la NASA [17] pour le Software Health Management (SWHM) pour détecter et diagnostiquer les défaillances d'origine

logicielle ou bien matérielle. Leur modèle se compose de différents types de nœuds diNCSets :

- nœud *Commande* C : ce nœud représente une vérité terrain et indique une commande, une action, un mode ou bien d'autres états connus. C'est un nœud observable;
- nœud *Sensor* S : ce nœud est observable aussi. Il représente, par exemple, les mesures obtenues de la surveillance du logiciel et du matériel;
- nœud *Health* H : ce nœud (non mesurable ou observable) représente l'état d'un composant ou d'un capteur. L'état de ce nœud est la sortie du modèle;
- nœud *Status* U : ce nœud représente l'état non observable d'un composant logiciel ou matériel ou l'état du sous-système.

Ils considèrent que seuls les nœuds C et S (entrées du RB) et le nœud H (sortie du RB) sont nécessaires dans leur modèle. Les autres nœuds sont utiles mais non nécessaires.

Przytula et Choi [18] ont développé un modèle bayésien pour le diagnostic et le pronostic qui a une structure en couches comme le montre la figure 6 :

- couche *components* : elle est constituée des nœuds composants ou de leurs modes de défaillances dont les états sont le résultat du diagnostic;
- couche *health observations* : elle est composée des nœuds d'observations de l'état des composants;
- couche *sub-systems* : elle regroupe les nœuds qui représentent l'état du sous-système;
- couche *usage* : ce sont des nœuds observables qui représentent l'usage du système dans le présent (dans le cas du diagnostic) et dans le futur (dans le cas du pronostic).

Une telle structure permet de réduire le nombre de paramètres nécessaires pour définir le réseau bayésien. L'objectif est de définir le diagnostic des composants à partir des observations et des usages.

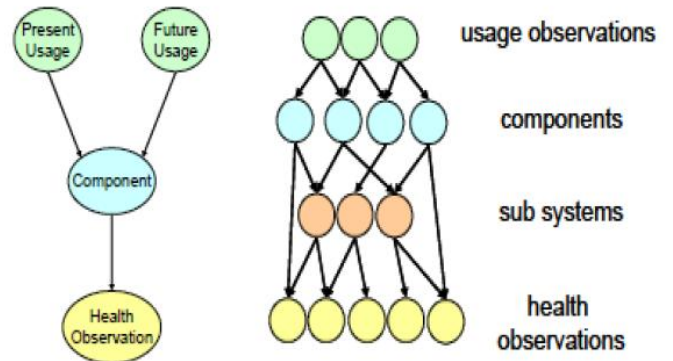


Fig. 6 : Modèle bayésien en couches de diagnostic [18]

La méthode de Przytula et Thompson [19] consiste à décomposer le système en sous-systèmes et à les modéliser séparément. Finalement, on intègre les sous-modèles en un seul modèle global. Pour construire un tel modèle il faut :

- décomposer le système en sous-systèmes
- définir les composants du système/sous-systèmes;
- définir les observations disponibles (tests, symptômes, messages d'erreur);
- définir une liste de défaillances des composants;
- faire correspondre les observations pertinentes aux défaillances.

Dans ce qui suit, nous présentons notre méthode bayésienne de diagnostic en nous basant sur les modèles cités précédemment.

IV. METHODE DE DIAGNOSTIC D'UN ROBOT MOBILE COMMUNICANT

Le système est composé d'un robot, équipé d'actionneurs et de capteurs, communiquant avec une station distante. Notre modélisation est le résultat de l'étude effectuée sur les défaillances du système commandé en réseau qui ont un impact sur la qualité de commande QoC du système en terme de stabilité et de temps de réponse, c'est-à-dire la capacité du contrôleur à amener le robot à la position d'équilibre via le réseau. La QoC est affectée par l'état du réseau, comme étant un composant du système global, en d'autres termes sa qualité de service QoS, et par l'état du robot. On divise le système global en trois sous-systèmes : une partie réseau, une partie opérative (actionneurs

et capteurs) et une partie commande. L'objectif de notre diagnostic est d'identifier la cause de dégradation de la QoC : soit une défaillance réseau, soit une défaillance au niveau du robot, soit une défaillance au niveau du contrôleur. Le réseau bayésien est construit avec une architecture en couches comme le modèle de Przytula et Choi [18].

A. Partie opérative

Le robot est équipé d'un ensemble de capteurs et d'actionneurs qui constituent la partie opérative du NCS. C'est la partie déterministe du système. On lui définit un nœud de type *sous-système* qui est *état du robot* qui indique l'état de cette partie comme le montre la figure 7. Les composants (nœuds composants) de cette partie sont : le moteur gauche,

le moteur droit et les capteurs d'obstacles. À chaque nœud composant, on associe un nœud observation qui a pour rôle de mesurer et surveiller l'état de santé de chacun. Le capteur est en état défaillant si les mesures du capteur restent bloquées à une certaine valeur ou transmet des valeurs erronées. Pour les actionneurs, le défaut moteur est le défaut sur la vitesse angulaire. Donc, l'état du moteur est observable par l'erreur de consigne qui ne doit pas dépasser un seuil sinon le moteur est dans un état défaillant. On définit un nœud usage "batterie" qui représente le mode d'utilisation de la batterie du robot. Si l'un des composants de la partie opérative tombe en panne, on doit le remplacer.

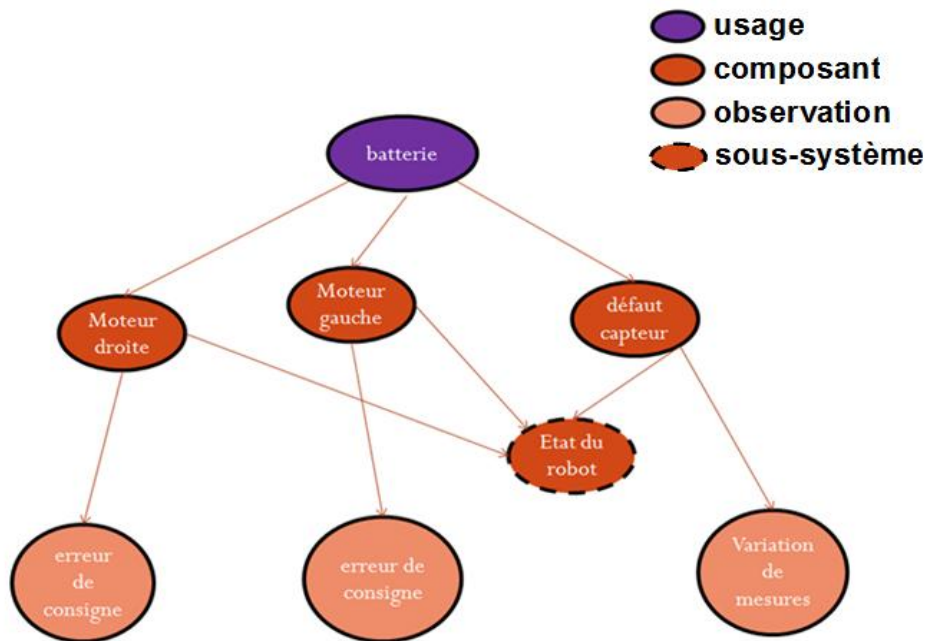


Fig. 7 : Modélisation bayésienne de la partie opérative

B. Partie commande

Le nœud QoC, qui représente l'état de la partie commande, indique la performance et l'état de l'application de contrôle-commande c'est-à-dire si le système est contrôlable via le réseau ou non. La QoC est influencée par l'état du robot (état de la partie opérative), l'état du réseau et par l'état du contrôleur comme le montre la figure 8. En effet, si les moteurs (les actionneurs) n'arrivent pas à

exécuter les commandes, on ne peut pas commander le robot.

Pour observer l'état de la QoC, on définit 3 nœuds d'observations :

- le nœud *stabilité* : ce nœud évalue la stabilité global du système le long de la trajectoire;
- le nœud *navigation* : ce nœud surveille la mission de navigation du robot (le robot doit atteindre la cible dans un temps fini);

- le nœud *évitement d'obstacle* : ce nœud surveille la mission d'évitement d'obstacle parce que le robot doit atteindre la cible en évitant l'obstacle.

Le nœud contrôleur est un nœud composant qui a une influence sur l'état de la QoC. Si le contrôleur est à faible capacité de calcul ou si son processeur (CPU) est chargé, on aura une dégradation de la QoC. On définit alors 2 nœuds *usage* pour le contrôleur :

- type de contrôleur : le contrôleur embarqué dans le robot est un contrôleur dégradé par contre le contrôleur débarqué sur la station est un contrôleur performant;
- charge CPU : c'est la charge utilisée par les applications (les ressources allouées aux applications) qui s'exécutent sur le processeur.

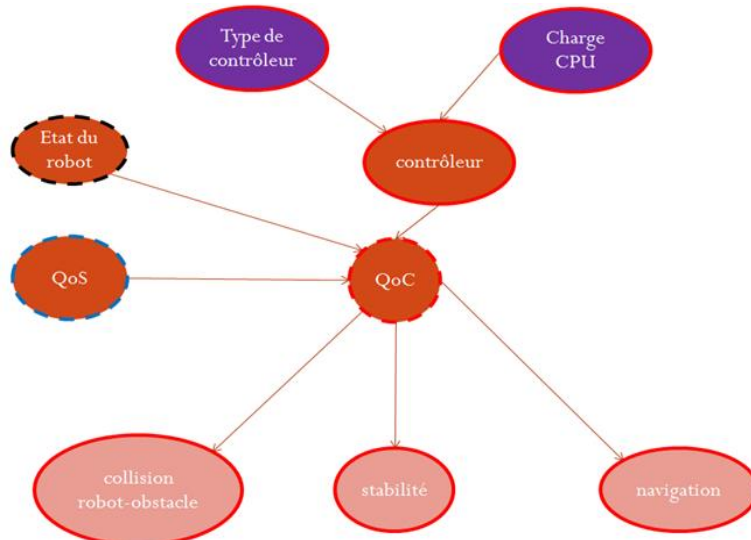


Fig. 8 : Modélisation bayésienne de la partie commande

C. Partie réseau

On considère le réseau comme un composant susceptible d'être défaillant comme les autres composants du système. Lorsque le réseau cesse d'assurer la QoS requise, il est considéré comme défaillant. Donc, l'état du réseau est représenté par l'état du nœud QoS qui est un nœud de type *sous-système* comme le montre la figure 9. La QoS désigne la capacité à fournir un support de communication conforme aux exigences en matière de temps et de bande passante et à garantir un niveau acceptable de pertes de paquets. Les paramètres influençant la QoS sont les pertes de paquets, le délai, la gigue et le débit. On a donc défini des défaillances observables par ces paramètres et qui ont une influence sur l'état du réseau, c'est-à-dire l'état de sa qualité de service. Pour l'information qui circule dans le réseau, on définit les types de défaillances suivants qui

peuvent lui arriver et provoquer un problème de transmission :

- perte de données : l'envoi des informations via un réseau entraîne des pertes à cause d'erreurs au niveau de la couche physique, de défaillance au niveau de la liaison ou de congestion du réseau. Pour les systèmes temps réel, un long retard est considéré comme une perte de l'information. L'état de ce nœud est surveillé par le nœud perte de paquets qui donne le taux de pertes de paquets à chaque envoi de messages.
- respect du délai : le délai dans un réseau wifi est aléatoire et variable. Le système fonctionne périodiquement et le non respect de la date limite d'arrivée (gigue ou bien long délai) des messages entraîne leurs pertes.

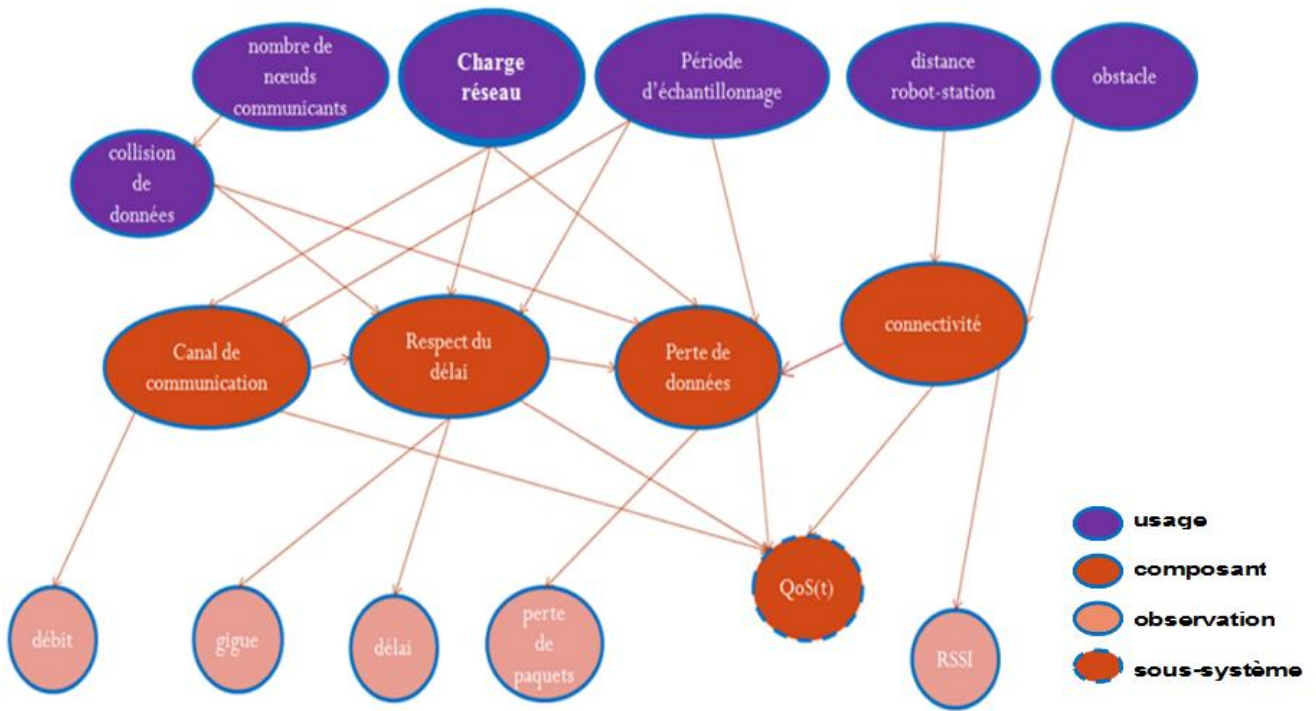


Fig. 9 : Modélisation bayésienne de la partie réseau

Un autre problème qui peut perturber la disponibilité du réseau est la limite de la bande passante. Si le canal de communication est saturé, l'envoi de messages n'est pas garanti, ce qui va entraîner une dégradation de la QoS. Le débit (qu'on définit comme nœud d'observation pour l'état du canal de communication) mesure la quantité d'informations envoyées par seconde, il quantifie la limite de la bande passante dans le réseau.

Une autre défaillance qui a un impact aussi sur la performance du réseau est la perte de connectivité. Si le robot s'éloigne de la station et sort de sa zone de couverture, il va perdre la connexion avec la station, ce qui va engendrer une perte de messages. On définit le nœud RSSI, qui mesure la puissance du signal reçu, comme nœud observation pour ce type de défaillance.

On définit des conditions d'utilisation du réseau par l'ensemble des nœuds de la couche *usage* :

- le nombre de nœuds communicants : il définit le nombre d'entités communicantes dans le réseau. En augmentant ce nombre, la probabilité d'avoir une collision augmente.

Si le nombre de nœuds augmente et la probabilité de collision augmente, le débit diminue, le retard augmente et le taux de pertes de paquets augmente aussi.

- la charge réseau : c'est la charge utilisée par les différentes applications qui partagent le réseau. Si le réseau devient surchargé, il perd ses performances, le retard augmente, le débit diminue et le taux de pertes de paquets augmente.
- la période d'échantillonnage : on définit une période d'échantillonnage pour la boucle principale de commande qui ne doit pas être inférieure à une limite, sinon le réseau va être surchargé et les messages envoyés vont être perdus.
- obstacle : ce nœud définit le milieu dans lequel le robot navigue, soit un milieu intérieur (indoor communication) c'est-à-dire un milieu avec la présence d'obstacle de natures différentes (bois, plastique, béton, métal...) ou à l'extérieur (outdoor communication).

- distance robot-station : cette distance définit la zone de couverture de la station. Si le robot dépasse cette zone, il perdra la connexion avec la station.

V. CONCLUSIONS

Les systèmes commandés en réseau sont des systèmes présentant des dysfonctionnements aléatoires dus à la présence du réseau de communication dans la boucle de commande. Les dysfonctionnements pour ce type de systèmes proviennent du réseau, de la partie opérative (le robot) qui exécute la commande ou du contrôleur qui est le calculateur des lois de commandes. Les réseaux bayésiens permettent un diagnostic probabiliste qui convient à la modélisation dans l'incertain. Le réseau bayésien développé présente les relations de causes à effets entre les différentes parties du système. Il sert à déterminer la cause de dégradation de la QoC, de la QoS et l'état interne du robot.

REFERENCES

- [1] C. Aubrun, D. Simon and Y. Q. Song, *Co-design Approaches for Dependable Networked Control Systems*, 1^{ère} édition, ISTE-WILEY, 2010.
- [2] R. Ghostine, *Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau*, thèse de doctorat. Université de Nancy, Institut National Polytechnique de Lorraine, 2008.
- [3] L. Zhang, H. Gao and O. Kaynak, *Network-Induced Constraints in Networked Control Systems*, IEEE Transactions on Industrial Informatics, Vol. 9, NO. 1, 2013.
- [4] A. Mechraoui, *Co-conception d'un système commandé en réseau sans fil à l'aide de réseaux bayésiens distribués*, thèse de doctorat, Université de Grenoble, Institut Polytechnique de Grenoble, 2010.
- [5] J. P. Hespanha, P. Naghshtabrizi and Y. Xu, *A survey of Recent Results in Networked Control Systems*, Proceedings of IEEE, 2007.
- [6] F. L. Lian, J. Moyne and D. Tilbury, *Network Design Consideration for Distributed Control Systems*, IEEE Transactions on Control Systems Technology, Vol. 10, NO. 2, 2002.
- [7] Y. Tipsuwan and M. Y. Chow, *Control methodologies in networked control systems*, Control engineering practice, vol. 11, no 10, p. 1099-1111, 2003.
- [8] J. R. Moyne and D. M. Tilbury, *The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics and Safety Data*, Proceedings of the IEEE, vol. 95, no 1, p. 29-47, 2007.
- [9] G. Habib, *Qualité de service et qualité de contrôle d'un système contrôlé en réseau sans fil : proposition d'une approche de co-conception appliquée au standard 802.11*, thèse de doctorat, Université Henri Poincaré Nancy I, 2010.
- [10] A. Cervin, D. Henriksson and M. Ohlin, *TrueTime 2.0 beta-Reference Manual*, Department of automatic control, Lund University, June 2010.
- [11] C. Berbra, *Diagnostic des systèmes embarqués en réseau : application à mini-drone hélicoptère*, thèse de doctorat, Université de Grenoble, Institut Polytechnique de Grenoble, 2009.
- [12] K. Medjaher, A. Mechraoui and N. Zerhouni, *Diagnostic et pronostic de défaillances par réseaux bayésiens*, 4^{èmes} Journées Francophones sur les Réseaux Bayésiens, JFRB'2008, Mai 2008.
- [13] P. Naïm, P. H. Wuillemin, P. Leray, O. Pourret and A. Becker, *Réseaux bayésiens*, 3^{ème} édition, Eyrolles, 2007.
- [14] P. Weber and C. Simon, *Réseaux bayésiens : un nouveau formalisme de modélisation pour la sûreté de fonctionnement*, AFNOR, BIVI Maître des risques, p1-16, 2013.
- [15] S. Bottone, D. Lee and M. Spivack, *Failure prediction and diagnosis for satellite monitoring system using bayesian network*, IEEE Military Communications Conference, MILCOM 2008, p. 1-7, 2008.
- [16] Y. Cheng, T. Xu and L. Yang, *Bayesian Network Based Fault Diagnosis and Maintenance for High-Speed Train Control Systems*, IEEE International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), p. 1753-1757, 2013.
- [17] J. Schumann, T. Mbaya and O. J. Mengshoel, *Software and System Health Management for Autonomous Robotics Missions*, Proc. of i-SAIRAS, 2012.
- [18] K. W. Przytula and A. Choi, *Reasoning Framework for Diagnosis and Prognosis*, IEEE Aerospace Conference, 2007.
- [19] K. W. Przytula and D. Thompson, *Construction of Bayesian Networks for Diagnosis*, IEEE Aerospace Conference Proceedings, 2000.
- [20] R. Isermann, *Model-based fault-detection and diagnosis—status and applications*, Annual Reviews in control, vol. 29, no 1, p. 71-85, 2005.